

汎用的能力を育成するための授業デザインを指向した
中学校・総合的な学習の時間の目標の分析的理解
—プログラミング活動を通じた目標達成の検討—

舟 生 日 出 男

『教育学論集』第70号

(2018 年 3 月)

汎用的能力を育成するための授業デザインを指向した 中学校・総合的な学習の時間の目標の分析的理解 ープログラミング活動を通した目標達成の検討ー

舟 生 日 出 男

1. はじめに

我々が住む社会は、加速度的に変化しつつある。内閣府（2017a）によれば、フィジカル空間にいる人々がサイバー空間にアクセスして情報を入手したり分析する現在の情報社会（Society 4.0）から、「サイバー空間（仮想空間）とフィジカル空間（現実空間）を高度に融合させたシステムにより、経済発展と社会的課題の解決を両立する、人間中心の社会（Society）」である Society 5.0 への変革が目指されている。確かに、このような変化の中で、AI（人工知能）やロボット、IoT（Internet of Things）の発展によって、人間の仕事のあり方も変わりつつある。例えば、メガバンクにおいても、将来を見越して人員を削減し、その削減分をデジタル化や AI の活用によって穴埋めすることが打ち出されている。今後はさらに、定式化できる処理は AI を中心に機械化されることは間違いないだろう。

そうした将来を見据えると、これからの生徒は、機械にはできないことを成し遂げる能力や、機械を活用したり、機械と協調して働くための能力を身に付けることが必須である。それらに関連する能力については、国立教育政策研究所（2016）によって「21 世紀に求められる資質・能力」としてまとめられており、その中で「深く考える（思考力）」として、「問題解決・発見」「論理的・批判的・創造的思考」「メタ認知・学び方の学び」が挙げられている。

このような背景を踏まえて、中学校の「総合的な学習の時間」では、教科横断的で汎用的な能力の育成が求められており、「21 世紀に求められる資質・能力」を構成する要素についても扱われている。しかし、学習指導要領で示されている目標や内容はやや抽象的であり、これらに基づいて指導計画を作成することは容易ではない。

ところで、上述の能力を育成する上で、プログラミング教育は重要な位置を占めると言える。なぜなら、プログラミングでは、対象の処理をアルゴリズム的に考えたり、粒度を考えながら課題を分割したり、それらの関係や構造についてシステムの捉えたりする力が求められるからである。さらに言えば、機械の活用や機械との協働において、プログラミングに関する知識やスキルは極めて重要である。

そこで本論文では、中学校の「総合的な学習の時間」の目標を分析的に捉えたとともに、中学校において、プログラミング教育が、技術・家庭における授業実践だけでは不十分であり、総合的な学習の時間においても扱うべきであること、それによって「総合的な学習の時間」の目標を十分に達成できることを示す。

2. 中学校「総合的な学習の時間」の目標の分析的理解

中学校の「総合的な学習の時間」の目標について、中学校学習指導要領（文部科学省 2017）の第4章・第1では、次のように示されている。

探究的な見方・考え方を働かせ、横断的・総合的な学習を行うことを通して、よりよく課題を解決し、自己の生き方を考えていくための資質・能力を次のとおり育成することを目指す。

- (1) 探究的な学習の過程において、課題の解決に必要な知識及び技能を身に付け、課題に関わる概念を形成し、探究的な学習のよさを理解するようにする。
- (2) 実社会や実生活の中から問いを見だし、自分で課題を立て、情報を集め、整理・分析して、まとめ・表現することができるようにする。
- (3) 探究的な学習に主体的・協働的に取り組むとともに、互いのよさを生かしながら、積極的に社会に参画しようとする態度を養う。

これらの3つの目標について、総合的な学習の時間の授業をデザインする上で、学習目標や評価の規準・基準の設定をより明確にできるよう、本論文では以下の9要素に分解して捉える。

(1) 知識・技能

- 1a. 探究的な学習の過程において、課題の解決に必要な知識及び技能を身に付ける **【探究的な学習を遂行するための知識・技能】**
- 1b. 課題に関わる概念を形成する **【関連づけによる概念形成】**
- 1c. 探究的な学習のよさを理解する **【探究的な学習の利点の理解】**

(2) 思考力・判断力・表現力等

- 2a. 実社会や実生活の中から問いを見だし、自分で課題を立てる **【問題発見・課題設定】**
- 2b. 情報を集め、整理・分析する **【情報の収集・整理・分析】**
- 2c. まとめ・表現する **【総括・表現】**

(3) 学びに向かう力・人間性等

- 3a. 探究的な学習に主体的・協働的に取り組む

【探究的学習における主体性・協働性】

3b. 互いのよさを生かす

【互恵的な協働】

3c. 積極的に社会に参画しようとする

【社会参画への志向】

このうち、1bの「課題に関わる概念を形成する」については、表現が曖昧である。黒上（2017）によれば、課題に関わる「概念」とは、知識と知識が関連づけられて構造化されたものである。生徒は、実社会・実世界の具体的な状況と結びついた形で、新しい知識と出会う。そのため、生徒自身が知識の意味や価値を実感しながら構造化できるように、学んだ事柄どうしの関連を常に検討させ、意味付けさせていく機会を設けることが重要である。

前述の「21世紀に求められる資質・能力」との関連を考えると、「問題解決・発見」はそのまま2aに対応している。「論理的・批判的・創造的思考」の内、「論理的・批判的（思考）」は「整理・分析する」ためには論理的に考えたり、批判的に捉える必要があることから2bに、「創造的思考」はまとめたり、表現する上で自分なりの工夫が必要となることから2cに対応していると言える。「メタ認知・学び方の学び」は、課題の解決に必要な知識及び技能を身に付ける中で、教師や他の生徒たちから助力を得ながら習得できると考えられることから、1aに対応していると言えるだろう。

3. 中学校でのプログラミング教育の問題

プログラミング教育は、中学校では2021年度から完全実施される。その内容について、「小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議」による「議論のとりまとめ」では、「中学校技術・家庭科技術分野の『情報に関する技術』において、計測・制御に関するプログラミングだけでなく、コンテンツに関するプログラミングを指導内容に盛り込むことによって、プログラミングに関する内容を倍増させること」が示されている（文部科学省 2016）。このように中学校については、計測・制御やコンテンツに関する「プログラミング＝コーディング」を指導することとされている。

一方で、その前段階の小学校におけるプログラミング教育については、「子供たちに、コンピュータに意図した処理を行うよう指示することができるということを体験させながら、将来どのような職業に就くとしても、時代を超えて普遍的に求められる力としての『プログラミング的思考』などを育むことであり、コーディングを覚えることが目的ではない」とされている（文部科学省 2016）。このように小学校では、プログラミング的思考の育成が中心である。

この「プログラミング的思考」については、「自分が意図する一連の活動を実現す

るために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」であると定義されている（文部科学省 2016）。例えば、画面上でキャラクターを、スタートからゴールまで障害物を避けながら移動させるようなプログラムで、「前進」「右に 90 度回転」「左に 90 度回転」「繰り返す」といった命令を組み合わせる上で、達成する手順を考える力である。また、目的を達成するプログラムは、冗長性にこだわらなければ、何通りでも考えることができる。そこで、冗長性を排して命令の数をできるだけ少なくしたり、他者がそのプログラムを見たときに、理解しやすくする工夫が求められる。「論理的に考えていく力」とは、このように考える力を指している。

学習指導要領における扱いを眺めると、小学校では、第 1 章総則の第 3 の 1 の (3) のイに「児童がプログラミングを体験しながら、コンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動」が掲げられ、算数、理科、総合的な学習の時間、のそれぞれにおける第 3 の「指導計画の作成と内容の取扱い」の中で「プログラミングを体験しながら論理的思考力を身に付けるための学習活動を行う場合には」と示されている。このように、小学校ではプログラミングを体験させることを通して、上述した論理的思考力を育成することが求められている。

一方の中学校ではプログラミングに関する扱いが少なく、技術・家庭の「D 情報の技術」(1)～(4)の内、「(2) 生活や社会における問題を、ネットワークを利用した双方向性のあるコンテンツのプログラミングによって解決する活動を通して、次の事項を身に付けることができるよう指導する」と「(3) 生活や社会における問題を、計測・制御のプログラミングによって解決する活動を通して、次の事項を身に付けることができるよう指導する」の 2 箇所で示されているにとどまる。ここでは「ネットワークを利用した双方向性のあるコンテンツ」と「計測・制御」の 2 点が触れられており、先述の「議論のとりまとめ」が示した内容と一致する。

これらの位置づけを踏まえると、小学校で育んだ「プログラミング的思考＝論理的に考えていく力」をベースに、中学校の技術・家庭において、「プログラミング＝コーディング」することが求められていると言える。

しかし、小学校においてプログラミング的思考力が十分に育成されている保証はない。多くの小学校では、2018 年度時点においてプログラミング教育を実施できる教員がほとんどいないこと、教材や機材が十分に揃っていないことから、手探りの状態からプログラミング教育を始めなければならないためである。

中学校においても機材、教材の問題は同様である。また、教員については逆に、技術を担当する教員はプログラミングに長けすぎていることがあり、レディネスが不十分な生徒とのギャップが大きい可能性がある。

これらの点を考慮すると、中学校で生徒のレディネスが十分ではないことに加え、

教員や教材、機材の問題についても、無視できない。従って、少なくとも当面の間は、技術・家庭の「D 情報の技術」で生徒にプログラミング=コーディングさせることは非現実的であり、何らかの手立てが必要であると言える。その1つとして、総合的な学習の時間において、プログラミングの活動を取り入れ、プログラミング的思考についての不足を補ったり、生徒の様子を見ながらプログラミング=コーディングの基礎について学ぶ場を設けるべきであると考ええる。

4. プログラミング的思考とコンピューショナル・シンキング

日本に比べて海外ではプログラミング教育が進んでおり、イギリスやオーストラリア、アメリカでは初等中等教育において全国レベルで実施されている（太田ほか2016）。

その内、イギリスでは、「2013年に従来の教科ICTに代わる新しい教科コンピューティング（Computing）のナショナルカリキュラムが告示され（DfE 2013）、2014年から実施され」ている。その中核は「コンピューショナル・シンキング（Computational Thinking, 計算論的思考）」である。コンピューショナル・シンキングは元々、Wing（2016）が提唱した概念であり、「Computational」な思考ではあるが、その対象はコンピュータを用いた問題解決に限定されるのではなく、一般的な問題解決も含んでいる。太田ほか（2016）はCAS（2015）を基に、英国の教科コンピューティングでのコンピューショナル・シンキングの概念を次のようにまとめている。

抽象化：問題を単純化するため、重要な部分は残し、不要な詳細は削除する。

デコンポジション：問題や事象をいくつかの部分に、理解や解決できるように分解する。

アルゴリズム的思考：問題を解決するための明確な手順で、同様の問題に共通して利用できるものである。

評価：アルゴリズム、システムや手順などの解決方法が正しいか、確認する過程である。

一般化：類似性からパターンを見つけて、それを予測、規則の作成、問題解決に使用する。

先述のプログラミング的思考をコンピューショナル・シンキングと対比して考えると、プログラミング的思考にはこれらの5つの内、「アルゴリズム的思考」と「評価」しか含まれていないことが分かる。その前段階となる「抽象化」と「デコンポジション」が正しく取り扱われていない場合、時間をかけてプログラミングに取り組ん

だとしても、プログラムを作る上で組み合わせる要素（制御、変数、関数、メソッドなど）について、生徒の理解が伴っていないことが多い。このような状態では、たとえプログラムを完成させたとしても、その意味も分からずに単に「こなした」だけにとどまり、興味関心も持たないままに終わってしまう。また、「一般化」が十分になされなければ、プログラミングの経験は特定のプログラムを作成したことだけに閉じられてしまう。そのため、プログラムが実社会の中のどのような場面で活用されているのか、プログラミングの知識・スキルを生活や仕事の場面でどのように活かせるのか、といったことを実感できないままとなる。

このことから、プログラミング教育では、コンピューショナル・シンキングの概念に基づいてプログラミングの思考を捉え直し、プログラミング活動の意味づけを十分に行うことが必要であると言える。

総合的な学習の時間の目標との関連では、抽象化とデコンポジション、アルゴリズムの思考は、問題の分析や分解、要素の取捨選択や再構成に関する活動であるから「2b. 情報の収集・整理・分析」に、評価は「21 世紀に求められる資質・能力」との関連で考えたように、メタ認知や学び方の学びに通じることから、「1a. 探究的な学習を遂行するための知識・技能」に、一般化は「1b. 関連づけによる概念形成」に対応していると捉えることができる。

5. 総合的な学習の時間の目標とプログラミング活動の困難性

プログラミング教育において、プログラミング活動を運営することには困難が伴う。その原因として、教師自身のプログラミング能力の低さも考えられるが、ここでは特に、1) 生徒の個人差が大きいこと、2) 解としてのプログラムが一意ではないことの2点について考察する。

まず1点目の、生徒の個人差が大きいことについて考える。数学においても同様の傾向が見られるが、理解や作業の進捗が生徒によって多様であり、授業時間内でその差に対応することは容易ではない。その解決のために、個別学習を導入し、ドリル型やチュートリアル型の教材を利用することも考えられる。しかし、このような方法では、プログラミングに対して苦手意識を持つ生徒にとっては、却って理解を妨げたり、遠ざける結果となりがねない。また、一見、理解しているように見えたとしても、浅い理解であったり、興味関心を伴っていないことも多いだろう。

むしろ、得意であったり興味を持っている生徒と、その逆に、不得意であったり興味を持てない生徒との協同学習を実施する方が高い学習効果を期待できる。不得意な生徒にとっては、得意な生徒から教えてもらったり助けてもらうことで、徐々に知識やスキル、興味関心を高めていくことができる。Vygotsky の「発達の最近接領域」に基づいて考えると、このような活動は理に適っていると言えるだろう。また、

得意な生徒にとっては教えたりアドバイスをすることは、必ずしも容易ではない。なぜなら、相手に応じてその内容をより易しくしたり、異なる角度から説明したり、例を用いるなど、工夫をしなければならないことが多いからである。そのように対話的に活動することを通して柔軟に「足場掛け」することができるようになる。

このように考えると、生徒の個人差を逆に利用することで、プログラミング活動を通して、相互に助け合い、補い合うことが必然的に発生することが分かる。総合的な学習の時間の目標との関連では、「3a. 探究的学習における主体性・協働性」と「3b. 互恵的な協働」に対応していると言える。

次に2点目の、解としてのプログラムが一意ではないことについて考える。先述した「画面上でキャラクターを、スタートからゴールまで障害物を避けながら移動させるプログラム」のような単純なプログラムであっても、その書き方は何通りもある。当然、「より良い」プログラムを設定することは可能であるが、その「良さ」について、いくつかの観点を設定することができる。

例えば、「できるだけ短い」という観点であれば、プログラムの可読性を度外視して、アクロバティックなプログラムを書いたり、汎用性を考えずに、その特定の状況だけに対応したプログラムを書くこともできる。一方で「手順が分かりやすい」という観点であれば、他者から見ても見やすく、後になって自身で見返しても理解しやすい、といった利点を伴う。また、このようなプログラムであれば、状況が変わった場面においても修正や拡張が容易であり、汎用性が高いと言える。

このように、プログラムを評価するためには何らかの観点を設定する必要があり、そのためには、クラスや学年、学校といったコミュニティの中での対話や合意形成が必須となる。このことは突き詰めると、実社会の中でどのように規定されているのかを考えたり、さらに言えば、自分達の考えを実社会に対して表明していくことにもつながるだろう。つまり、個人や一部の中でプログラミングを考えるのではなく、より大きな枠組みの中で考えなければならないと言える。実際に、Scratch などでは、自作したプログラムを共有し、評価し合う仕組みがあり、このことは過言ではない。総合的な学習の時間の目標との関連では、「3c. 社会参画への志向」に対応していると言える。

6. 総合的な学習の時間の目標とプログラミング活動との関連

3章と4章で考察した、総合的な学習の時間の目標とコンピューショナル・シンキングやプログラミング活動の困難性との関連についてまとめたものを表1に示す。この表から、目標の9要素の内、6要素をプログラミング教育によってカバーできることが分かる。その一方で3要素が残されており、これらについて補足する。

まず「1c. 探究的な学習の利点の理解」については、他の生徒との協働を通して

「3a. 探究的な学習における主体性・協働性」を達成する中で、探究的な学習の利点を理解できると考える。

次に「2a. 問題発見・課題設定」については、実際の授業では、生徒に幅広い対象からの課題選択を許すというよりも、1つの課題を与えるか、少数の課題から選択させるかのどちらかとなるのが現実的であろう。小中一貫校などで長期にわたって実践を積み上げているような場合は自由な課題選択も考えられるが、多くの場合、現状では困難である。

最後に「2c. 総括・表現」については、プログラミング活動では、作成したプログラムを発表したり、相互に評価し合うフェーズを含むのが普通である。ただし、単なる発表や主観的な評価に終わらずに、予め合意した観点に基づいた評価活動が重要である。

以上を整理すると、プログラミング活動によって「2a. 問題発見・課題設定」を除く8要素について、総合的な学習の時間の目標をカバーできると言える。

表1 総合的な学習の時間の目標とコンピューショナル・シンキング、プログラミング活動の困難性との関連

総合的な学習の時間の目標	コンピューショナル・シンキング	プログラミング活動の困難性
1a. 探究的な学習を遂行するための知識・技能	評価	
1b. 関連づけによる概念形成	一般化	
1c. 探究的な学習の利点の理解		
2a. 問題発見・課題設定		
2b. 情報の収集・整理・分析	抽象化 デコンポジション アルゴリズム的思考	
2c. 総括・表現		
3a. 探究的な学習における主体性・協働性		生徒の個人差が大きいこと
3b. 互恵的な協働		
3c. 社会参画への志向		解としてのプログラムが一意ではないこと

7. まとめ

本論文では、中学校の「総合的な学習の時間」の目標を分析的に捉えて9要素に分割した。その上で、中学校において、プログラミング教育が、技術・家庭における授業実践だけでは不十分であり、総合的な学習の時間においても扱うべきであることを論じた。

そして、コンピューショナル・シンキングに基づいて捉え直したプログラミング的思考を育成することは、総合的な学習の時間の目標達成にも通じること、プログラミング活動における困難性はむしろ、生徒の協働や対話を必然的に生じさせることを示し、プログラミング活動によって、「総合的な学習の時間」の目標を十分に達成できることを述べた。

今後の課題として、生徒の様々なレディネスごとに学習活動をデザインし、教材を開発することや、教師のプログラミング力やプログラミング活動のデザイン力を高めるためのワークショップを企画することを考えている。

引用・参考文献

Computing At School (CAS) “CAS Computing Progression Pathways KS1 (Y1) to KS3 (Y9) by topic,” <http://community.computingatschool.org.uk/resources/1692>, November 2015 (閲覧日 2018 年 1 月 6 日)

Department for Education (DfE) “National curriculum in England: computing programmes of study,” <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>, September 2013 (閲覧日 2018 年 1 月 6 日)

国立教育政策研究所編『資質・能力 理論編』東洋館出版社, 2016 年 1 月

黒上晴夫「1 章 1 資質・能力の三つの柱と目標の改訂」(田村学編著『新学習指導要領の展開 総合的な学習の時間編』明治図書出版, 2017 年 11 月, pp.16-21.)

文部科学省「小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)」http://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm, 2016 (閲覧日 2018 年 1 月 6 日)

内閣府「Society 5.0」http://www8.cao.go.jp/cstp/society5_0/, 2017a (閲覧日 2018 年 1 月 6 日)

内閣府「Society 5.0 それは、いつもの毎日にやってくる、半歩先の未来」<https://www.gov-online.go.jp/cam/s5/>, 2017b (閲覧日 2018 年 1 月 6 日)

太田剛, 森本容介, 加藤浩「諸外国のプログラミング教育を含む情報教育カリキュラ

ムに関する調査－英国，オーストラリア，米国を中心として－」日本教育工学会論文誌 40(3), 2016 年 12 月, pp.197-208.

Wing, J. M. “Computational Thinking” *Communication of the ACM* 49(3), March 2006, pp.33-35. (中島秀之翻訳「計算論的思考」情報処理 56(6), 2015 年 6 月, pp.584-587.)

Decomposition of the Goal of the Period for Integrated Studies in Junior High School to Design of Learning Activities to Foster Students' Generic Competence

—Discussion on Achieving the Goal through Programming Activities—

Hideo FUNAOI

Abstract

In this paper, the author divided analytically the goal of the Period for Integrated Studies, which are printed in the National Curriculum Standards for junior high school, into nine elements and related them to revised programming thinking based on computational thinking and two difficulties in programming activities.

Further, the author discussed that programming education at junior high school is insufficient by only Technology and Home Economics classes and should also be handled in the Period for Integrated Studies, and concluded that programming activities can achieve the goal of the Period for Integrated Studies sufficiently.